University of Rajshahi

Department of Computer Science and Engineering

B.Sc. (Engg) Part-2, Odd Semester Examination 2022

CSE2122 (Data Structure Lab) for Affiliated Colleges

Full Marks: 17.5 **Time: 3 Hours**

1. Given an integer array, find the maximum product of two integers in it. For example, consider array A with {-10, -3, 5, 6, -2}. The maximum product is the (-10, -3) or (5, 6) pair.

2. Write a program to evaluate an arithmetic expression written in postfix notation. The postfix notation consists of single letter, single digit and arithmetic operators (+, -, *, /, ^). If the expression contains a letter, you have to ask the value (single digit) for that letter.

Input	ab2*c3^/-	Output	-1
	Enter the value of a: 3 Enter the value of b: 2 Enter the value of c: 1		

3. Find out the shortest path of a Weighted Graph G with m nodes $V_1, V_2, \ldots, V_m$ and weight of each edge is w(e) using Warshall's Algorithm.

Example: Suppose $W=\begin{pmatrix} 3 & 5 & 0 & 0 \\ 3 & 0 & 0 & 7 \\ 2 & 5 & 3 & 4 \\ 3 & 2 & 4 & 0 \end{pmatrix}$ then the shortest path matrix is $\begin{pmatrix} 3 & 5 & 16 & 12 \\ 3 & 8 & 11 & 7 \\ 2 & 5 & 3 & 4 \\ 3 & 2 & 4 & 8 \end{pmatrix}$

University of Rajshahi

Department of Computer Science and Engineering

Part- II, Semester – Odd, Final Exam – 2022

Course: CSE2142 (Writing Professional Code)

Please write the following code.

```
//File Name StudentList.java
import java.io.*;
import java.text.*;
import java.util.*;
public class StudentList {
public static void main(String[] args) {
//      Check arguments
        if(args[0].equals("a")) {
            System.out.println("Loading data ...");
            try {
            BufferedReader s = new BufferedReader(
                new InputStreamReader(
                        new FileInputStream("students.txt")));
             String r = s.readLine(); String i[] = r.split(",");
             for(String j : i) { System.out.println(j); }
            } catch (Exception e){}
            System.out.println("Data Loaded.");
        }
        else if(args[0].equals("r"))
        {
            System.out.println("Loading data ...");
            try {
            BufferedReader s = new BufferedReader(
                    new InputStreamReader(
                            new FileInputStream("students.txt")));
            String r = s.readLine(); System.out.println(r);
            String i[] = r.split(",");
            Random x = new Random();
                int y = x.nextInt();
                    System.out.println(i[y]);
            } catch (Exception e){}
            System.out.println("Data Loaded.");
        }
```

```
else if(args[0].contains("+")){
   System.out.println("Loading data ...");
   try {
   BufferedWriter s = new BufferedWriter(
         new FileWriter("students.txt", true));
   String t = args[0].substring(1);
     Date d = new Date();
     String df = "dd/mm/yyyy-hh:mm:ss a";
     DateFormat dateFormat = new SimpleDateFormat(df);
     String fd= dateFormat.format(d);
    s.write(", "+t+"\nList last updated on "+fd);
    s.close();
    } catch (Exception e){}
    System.out.println("Data Loaded.");
 }
 else if(args[0].contains("?"))
 {
    System.out.println("Loading data ...");
    try {
    BufferedReader s = new BufferedReader(
          new InputStreamReader(
                new FileInputStream("students.txt")));
    String r = s.readLine();
    String i[] = r.split(",");
    boolean done = false;
    String t = args[0].substring(1);
    for(int idx = 0; idx<i.length && !done; idx++) {
        if(i[idx].equals(t)) {
           System.out.println("We found it!");
              done=true;

        }
    }
    } catch (Exception e){}
    System.out.println("Data Loaded.");
 }
 else if(args[0].contains("c"))
 {
    System.out.println("Loading data ...");
    try {
        BufferedReader s = new BufferedReader(
              new InputStreamReader(
                    new FileInputStream("students.txt")));
        String D = s.readLine();
        char a[] = D.toCharArray();
        boolean in_word = false;
        int count=0;
        for(char c:a) {
            if(c ==' ')
            {
                if (!in_word) { count++; in_word =true;   }
                else { in_word=false;}
            }
         }
         System.out.println(count +" word(s) found " + a.length);
         } catch (Exception e){}
         System.out.println("Data Loaded.");
       }
    }
 }
```

Data File name

```
students.txt
```

File Contents (Initial Stage)

```
Student1, Student2, Student3, Student4
```

Data file and java file should be in same location.

Steps #0 (Initial Stage as in question. Check the output whether it matches with the output given here. If your output doesn't match, please correct your program so that it produce exactly the same output given here.)

Run #1 `$java StudentList a`

```
Loading data ...
Student1
Student2
Student3
Student4
Data Loaded.
```

Run #2 `$java StudentList r`

```
Loading data ...
Student3
Data Loaded.
```

Run #3 `$java StudentList r`

```
Loading data ...
Student1
Data Loaded.
```

Run #4 `$java StudentList c`

```
Loading data ...
2 word(s) found
Data Loaded.
```

Run #5 `$java StudentList ?Student1`

```
Loading data ...
We found it!
Data Loaded.
```

Run #6 `$java StudentList +Another`

```
Loading data ...
Data Loaded.
```

File Contents (After running)

```
Student1, Student2, Student3,
Student4 ,Another
List last updated on 2019-07-25
2:10:58 PM
```

In case you write the code by yourself instead of downloading, you need to initialize a git repository on your project. You should create git branch for every step with the name of the step, work on that particular step, commit the code with appropriate commit message and finally merge your branch to the master. For better understanding, write the commit message same as the task on each steps. *And check every time you change something, you didn't break anything by going through Run #1 to Run #6* and it matches the output.

STEP #1.Update code style for better consistency.

STEP #2.Application now terminates early if the number of arguments passed into it is wrong, fix it.

STEP #3.Makes improvements to variable names

STEP #4.Remove Temporary variables

STEP #5.Simplifies the logic behind the count operation 8

STEP #6.Adds handling for case when user enters invalid arguments 9

STEP #7.Add more comments and makes more naming improvements. 10